

# SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

## [METHOD FOR GENERATING COMMANDS FOR NETWORK CONTROLLER MODULES OF PERIPHERAL DEVICES]

### Background of Invention

[0001] *BACKGROUND FIELD OF INVENTION*

[0002] This invention relates to electrical control communication systems and, more particularly, to a command system for data transmission between control units arranged in a network.

[0003] *BACKGROUND PRIOR ART*

[0004] Many commercially available products provide control and communications in a network environment. These products range from very expensive, elaborate systems, to simple systems having little intelligence. The present invention is directed towards providing a system having a relatively large amount of intelligence and computational power but at a low cost.

[0005] One commercially available system (i.e., "X-10") offers control, e.g., between a light switch and a light source. When the light switch is operated, a code pattern is transmitted over the power lines to a receiver at the light. The code pattern is transmitted twice, once in its true form and once in its complementary form. When the code is received by the receiver, it is interpreted and thereby used to control the light. Mechanical addressing means are needed to allow the transmitter at the switch to communicate with the specific receiver at the light.

[0006] The command protocol disclosed herein provides great flexibility. The

communication between cells is optimized for the performance of the functions assigned to units, rather than for transmission of data unrelated to the control function of the network. For this reason, in general, packets carrying messages are relatively short compared to Ethernet, Arpa, Apple Talk, X-25 and many other broadband and data communication systems.

[0007] Through the proposed command system, device-generated network traffic can be minimized, thus allowing several units to use one network without reducing communication quality or increasing the wait period before information can be transmitted. This approach is so versatile that it supports extremely simple communication networks, comparable in capacity and reliability to other existing network systems, without any needed modification.

## Summary of Invention

[0008] The present invention discloses a method to generate commands for control device intercommunication across a network. Through these commands, interdevice information transmission and exchange is made possible. In addition, the present method supports the transmission of requests of actions to be executed by devices across the network. The command format is easy to implement, and uses very few device resources: a very important issue for the implementation of low-cost networks. The versatility of this method allows transmission of both information and requests of actions to be performed by other devices. The versatility of implementation of the commands allows the use of this system in different types of device networks, where there may be one master and several slaves, or several masters and several slaves, or where dynamic mode setting can be used as necessary.

[0009] OBJECTS AND ADVANTAGES OF INVENTION

[0010]

Accordingly, several objects of the present invention are:—To provide a format for data structure processing that allows communication among control devices in a network;—To provide a data structure that can be easily decoded by network devices, which consumes very few resources, and reduces implementation costs;—To provide a standard format that allows the inclusion of new commands in the command set already used in a network, maintaining compatibility with devices that do not have the

new complete set of commands;—To provide a command system that permits reliable critical data transmission, such as device configuration parameters and network-controller system variables.

[0011] In addition, several advantages of the present invention are:—The use of very few network device resources, thus reducing implementation costs and allowing the use of low-capacity and low-resource controllers for computations and communication handling;—Reduced amount of fields to transmit by information packet, reducing the channel's busy-time;—It is not necessary to transmit preambles to establish synchronization between the command transmitter and receiver;—The possibility of using different packets lengths with no need to modify the information processing algorithm;—Ease of information packet delivery, with no need of using packet switching devices nor routers;—The versatility of functions provided to the network units in which it is implemented. The functions may handle the distribution of packets within subnets, handle the group directions, and handle group functions, among others.

## Brief Description of Drawings

- [0012] Fig. 1 Fixed-function Master/Slaves Network
- [0013] Fig. 2 Dynamic-function Master/Slaves Network
- [0014] Fig. 3 Stand-by unit in the Network
- [0015] Fig. 4 Master to Slave communication
- [0016] Fig. 5 Remote unit intercommunication
- [0017] Fig. 6 Communication Packet
- [0018] Fig. 7 Communication flow
- [0019] Fig. 8 Main Communication algorithm
- [0020] *REFERENCE NUMERALS*
- [0021] 2 Network bus.

- [0022] 4 Master unit.
- [0023] 6 Slave unit.
- [0024] 8 Dynamic-mode Master unit.
- [0025] 10 Dynamic-mode Slave unit.
- [0026] 12 Stand-by unit.
- [0027] 14 Communication packets.
- [0028] 16 Acknowledge packets.

## Detailed Description

[0029] DETAILED DESCRIPTION OF DRAWINGS

[0030] *Fig. 1 – Fixed function Master/Slaves Network* Let there be a basic device network comprising a master unit 4 and N remote units 6, each having its own network address that uniquely identifies it in the network. Interconnection of units exists by means of bus 2. In this example, the mode of each unit is fixed. The masters will always be masters, and slaves will always be slaves.

[0031] *Fig. 2 – Dynamic function Master/Slaves Network* The device network may comprise units in dynamic mode. These units may switch between master-mode and slave-mode as necessary. When a unit switches to master-mode, the rest of the units must switch to slave-mode. The master units 8 connect to slave units 10 through bus 2.

[0032] *Fig. 3 Stand-by unit in the Network* In addition to master- and slave-modes, it is possible to set units to operate in stand-by mode 12. Stand-by units act as hidden slave units, listening to the network without sending data or acknowledges out to it.

[0033] *Fig. 4 Master to Slave communication* In the communication process, a master unit 8 sends an information packet 14 through the bus 2 to a slave 10. An acknowledge bit 16 is received from the slave 10 for each received byte. Packet 14 contains the address of the destination unit (i.e., the unit the packet must reach).

[0034] Fig. 5 Remote unit intercommunication When units have the capacity of dynamic mode, any unit may switch to master-mode 8 while the others remain in slave-mode 10. This way, any unit may send information to any other unit connected to the same bus 2 using the same communication and packet format.

[0035] Fig. 6 Communication Packet Information is encapsulated in packets for transmission. In general, these comprise four fields, each consisting of several bytes. The first field corresponds to the packet's destination address, and specifies the remote unit that must ultimately receive the packet. The second field corresponds to the command that specifies the function of the information contained in the packet. This command may indicate that the receiving unit must execute a specific action, process the packet's data in a specific manner, or generate a reply with information contained in the remote unit (i.e., request/response). The third field corresponds to data. This data may originate in the master unit and be sent to a slave unit, or be contained in a slave unit and be requested by the master unit, or a combination of both, where both master and slave units supply information. This is a bi-directional field, and its nature is determined by the command contained in the second field. The fourth field corresponds to information for communication verification or error detection. It may contain CRC, checksum, or other value.

[0036] Fig. 7 Communication flow In the communication process, the master unit 4 sends information bytes, and the slave unit 6 confirms reception of a packet by sending acknowledge (i.e., ACK) bits back to the master unit 4. The communication at any moment is bi-directional. This way, communication errors may be detected during the process, without having to wait until the end of the process.

[0037] Fig. 8 Main Communication algorithm The communication algorithm of remote units with capacity of dynamic-mode comprises several basic blocks. When a new device enters the network, it is set to slave-receiver mode so that it can listen to the bus. If it needs to transmit information, it is set to master-transmitter. In this case, it generates an information packet, verifies that the bus is available, and checks the bus's state of arbitration to avoid collisions. Then it transmits the packet. If a call is received, the unit receives the packet. If an error in the transmission process is detected, a unit may reattempt packet transmission. The entire process is always

checked by a time clock. This ensures that the system is not locked in a loop, generating a jump to the beginning of the communication algorithm.

[0038]      *OPERATION OF INVENTION* A master 4 is the device providing the signaling necessary to transmit information through the network. A slave 6 is the device receiving the information from the network, using the signaling present in the network bus 2. In case of a network in which the communication modules mode is defined so that the central module is the master 4 and the remote units are slaves 6, communication is established only when the master initiates it. If a remote slave unit 6 must transmit information to its master, it must wait for the master's call. In case of a network in which the communication modules mode is interchangeable between masters and slaves, remote units have the possibility of addressing the central device and requesting attention from it.

[0039]      In case of multimastering, a bus arbitration mechanism (i.e., bus arbiter) handles collisions when two or more masters 8 try to access the bus 2 simultaneously. It is possible to implement said bus arbiter through a dedicated bus arbitration device. However, this is not necessary. If desired, the master units 8 may compare the state of their outputs against the actual state of the bus 2. When a difference between the master's output and the state of the bus occurs, it follows that a collision is taking place (i.e., two or more masters are trying to transmit simultaneously). Then, the masters that detected the difference surrender access to the bus whereas the master whose output corresponded to the state of the bus gains access to the bus and continues transmission.

[0040]      Through multimastering, it is possible to implement a network in which both remote and local devices may exchange information among them without the intervention of an intermediary device or a device with higher hierarchy. Through a multimaster network, it is possible to delegate administrative tasks to lower hierarchy network devices, while higher hierarchy devices carry out their tasks. The administrator device may be in charge of handling network messages of other devices, taking care of its requests and resolving conflicts of local significance within the network. Important messages can be forwarded to the higher hierarchy device for processing.

[0041] The generic format of the commands permits the implementation of this type of network structure since it contemplates addressing and data structures for information flow between devices regardless of hierarchy. Packets contain an address field, a command field, a data field and an error correction field. The address field specifies the recipient's network address. The command field specifies actions to be performed by the addressed remote device. The data field contains information associated with the command. This data may be information that the remote device requires, or it may be data to be sent to the master unit by the remote device (i.e., data request).

[0042] The error correction field contains information to used to check for errors in packet transmission. The error correction field may contain a checksum, calculated in the master unit from all bytes to be transmitted, and in the slave unit from all the bytes received from the network. Through this checksum, it is possible to check for errors in the received information. It is advisable to use the destination device's address to generate the value of the checksum. If the checksum computed at the destination device does not match the checksum included in the packet, the packet is discarded.

[0043] It is not necessary to implement the complete structure of these format specifications. It is possible to establish communications in which a simplified format is used, consisting of an address field, a command field, and a bi-directional data field. The error correction field may be omitted if the underlying communication bus 2 is considered sufficiently (i.e., highly) reliable. In order to guarantee system stability in cases where the simplified communications format has been used, it is possible to implement a set of "enabling" configuration command that warn remote devices that the following commands contain configuration information. If a remote unit receives a configuration command without first having received the enabling configuration command, the configuration command is discarded. This mechanism is quite useful in networks in which low information flow on the bus is highly important because remote unit configuration happens very infrequently.

[0044] The communication process is performed by sending information in byte segments and receiving acknowledge replies indicating the byte reception. Basically,

bytes are sent and an acknowledge bit is received.

[0045] Communication starts with the transmission of the destination device's network address. Next, an acknowledge bit is expected. Once received, the following byte of information is transmitted, namely, the byte specifying the action that should be taken about the current communication process. In case of the typical process, this byte specifies the command to be executed at the remote unit. Next, a second acknowledge bit is expected. After receiving an acknowledge bit, the command data block is transmitted. This data can be sent from the master and can be read from the remote unit in the same packet (bi-directional communication), or in a single direction as in the case of a reading or configuration command.

[0046] The basic device communication algorithm is illustrated on the Fig 8. Communication starts by configuring a device as a slave-receiver unit, which receives events originating in the network. If a packet must be transmitted, the unit prepares a buffer containing the information, checks the availability of the bus, and switches its mode to master-transmitter mode by itself, following the communication algorithm. While transmission is taking place, the bus access state (i.e., arbitration) is checked. If bus access is lost due to conflict, transmission is reattempted. If bus access is gained, corresponding acknowledges are expected for each transmitted field. When communication ends successfully, the process concludes and the device returns to the start state of the algorithm. However, if there were transmission problems, transmission is reattempted. The number of retries depends on the application and specific programming needs.

[0047] In case a network call is received during the wait-cycle of packet transmission (i.e., when acknowledges are expected), the packet is received and a corresponding acknowledge is sent out. If errors occur, a NO\_ACK packet (i.e., the binary complement of an acknowledge packet) is sent to the master unit indicating that an error occurred. Then, the unit returns to the start of the network listening process.

[0048]

[0049]

[0050] The command protocol disclosed herein provides great flexibility. The



communication between cells is optimized for the performance of the functions assigned to units, rather than for transmission of data unrelated to the control function of the network.

[0051] Through the proposed command system, device-generated network traffic can be minimized, thus allowing several units to use one network without reducing communication quality or increasing the wait period before information can be transmitted. This approach is so versatile that it supports extremely simple communication networks, comparable in capacity and reliability to other existing network systems, without any needed modification.

[0052] The versatility of implementation of the commands allows the use of this system in different types of device networks, where there may be one master and several slaves, or several masters and several slaves, or where dynamic mode setting can be used as necessary.

[0053] In addition, several advantages of the present invention are:—The use of very few network device resources, thus reducing implementation costs and allowing the use of low-capacity and low-resource controllers for computations and communication handling;—Reduced amount of fields to transmit by information packet, reducing the channel's busy-time;—It is not necessary to transmit preambles to establish synchronization between the command transmitter and receiver;—The possibility of using different packets lengths with no need to modify the information processing algorithm;—Ease of information packet delivery, with no need of using packet switching devices nor routers;—The versatility of functions provided to the network units in which it is implemented. The functions may handle the distribution of packets within subnets, handle the group directions, and handle group functions, among others.

[0054] While our above description contains many specificities, these should not be construed as limitations to the scope of the invention, but rather as an exemplification of one preferred embodiment thereof. Obviously, modifications and alterations will occur to others upon a reading and understanding of this specification such as, for example, a command set which implements a stronger acknowledge format, where the acknowledge packet contains the communication status so the transmitter can

